

POPULAR Computing WEEKLY

9 September 1982 Vol 1 No 21

35p

Exclusive

We take the lid off
the Jupiter Ace

Play Simon on your BBC

Spectrum stock control

Vic relocation

Cover Story:
Pteragon on BBC

Whizzkid '82
Win a Dragon 32 & £2000
of advertising

POPULAR Computing WEEKLY

The Team

Editor

Brendon Gore

Reporter

David Kelly [01-930 3271]

Sub-editor

Ninette Sharp

Editorial Secretary

Fiona McCormick

Advertisement Manager

David Lake [01-839 2846]

Advertisement Executive

Alastair Macintosh [01-930 3840]

Managing Editor

Duncan Scot

Publishing Director

Jenny Ireland

Popular Computing Weekly,
Hobhouse Court, 19 Whitcomb Street,
London WC2
Telephone: 01-839 6835

Published by Sunshine Publications Ltd.

Typesetting, origination and printing by
Chesham Press, Chesham, Bucks

Distributed by S M Distribution
London SW9. 01-274 8611. Telex: 261643

© Sunshine Publications Ltd 1982

Subscriptions

You can have Popular Computing Weekly sent to your home: the subscription rate is £19.95 per year, for addresses in the UK, £37.40 overseas.

How to submit articles

Articles which are submitted for publication should not be more than 1000 words long.

All submissions should be typed and a double space should be left between each line.

Programs should, whenever possible, be computer printed.

At present we cannot guarantee to return every submitted article, so please keep a copy.

Accuracy

Popular Computing Weekly cannot accept any responsibility for any errors in programs we publish, although we will always try our best to make sure programs work.

This Week



Cover illustration by Ian Craig

News	5
Commodore Max for Christmas.	
Letters	7
Spectrum bugs wanted.	
Pteragon	8
A new game for BBC by A Snell.	
Street Life	11
David Kelly looks at The Pike.	
Reviews	12
Boris Allan presents the first review of the Jupiter Ace.	
Open Forum	14
Five and a half pages of your programs.	
Whizz kid '82	19
Win a Dragon 32.	
Programming	20
Vic relocation.	
Spectrum	21
Stock control by J Reynolds.	
Sound & vision	22
Play Simon on BBC.	
Peek & poke	23
Your questions answered.	
Competition	26
Puzzle, Arthur.	

Editorial

The Jupiter Ace, launched last month, is a new kind of low cost micro. Designed by Richard Altwasser and Steve Vickers, the team responsible for Sinclair's ZX Spectrum, the Jupiter Ace consists of a Z80A microprocessor complete with 8K ROM, 3K RAM and 32 x 24 display.

But, while the hardware is fairly standard, the software is decidedly different. The Jupiter Ace is the first micro to use Forth. Almost every other micro on the market uses some form of Basic.

Forth was invented by Charles Moore and Elizabeth Rutherford at the US National Radio Astronomy Observatory in the early 1970s. It is much faster than Basic and takes up far less memory. Supporters of Forth also claim it is easier to learn than Basic.

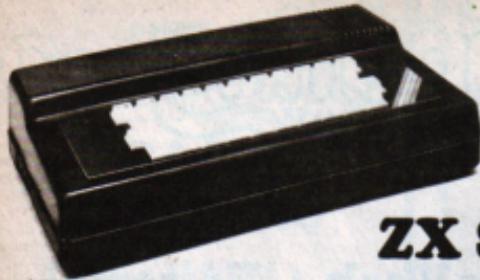
Calculator enthusiasts, who should be familiar with Reverse Polish Notation, will find little difficulty in coming to grips with the Jupiter Ace. Micro users, who have already learnt to program in Basic, may find it harder to deal with Forth.

Whatever your views on Forth, Altwasser and Vickers should be congratulated for trying something new. I just wish the Jupiter Ace was colour instead of black and white.

Next Week



Can you save your city from the mutant bees? Find out in Swarm, a new game for Spectrum.



New From Fuller
FD System for the

ZX SPECTRUM

£39.95

+ £2.50 p & p.

Professional Keyboard & Case —

This unit has the same high standard as our ZX81 unit.

Tough A.B.S. Plastic case encloses our Keyboard, the Spectrum Printed Circuit Board and the Power Supply.

Our own Power supply is available:- 9 volts DC at 2 amps.

Mains either 110v or 240v AC at £5.95 + 80p. p & p.

The Keyboard has 42 keys with all the spectrum functions printed onto them, the full travel key switches have gold plated contacts and a guaranteed life of 10^6 operations.

INSTALLATION - Simply unscrew the ZX printed circuit board from its case and screw it into the FD case, plug in the keyboard and that's it. No technical know how or soldering required, the built unit is tested and comes with a money back guarantee.

Spectrum Keyboard and Case Kit £33.95

Our Mother Board for the spectrum has 2 slots at £15.95 or 3 slots at £19.95, this unit also fits inside the case. p & p 80p.

SPECTRUM SOUND AMPLIFIER £5.95 + 80p p & p.

Complete with leads, volume control and loud speaker in tough ABS Plastic case measuring 5" x 3" x 1" just plugs into your spectrum MIC input.



First Anniversary Offer

The FD System is now one year old and Fuller are celebrating with this amazing offer on the FD42 Professional Keyboard and Case.

Makes an ideal Christmas present to expand the new low priced Sinclair ZX81. Or why not buy a new ZX81 based system directly from us, consisting of ZX81, FD42 keyboard and case with power supply and reset switch, leads and manual £69.95 + £2.50 p & p

FD42 Keyboard and Case Kit £24.95 + £2.50 p & p

FD42 Keyboard kit £14.95 80p p & p

£29.95

+ £2.50 p & p

STAR TREK FOR ZX 16K SPECTRUM

Play this popular adventure game on your Spectrum with ship display and sound £5.00 + 50p p & p

GUARANTEED 14 DAYS DELIVERY FROM RECEIPT OF ORDER, OR CALL TO THE ZX CENTRE,

Mail to **FULLER MICRO SYSTEMS**,

The ZX Centre, Sweeting Street, Liverpool 2. England, U.K.

Please Supply:-

Name

Address

FULLER FD SYSTEM

SAE for more details — Enquiries: Tel. 051-236 6109

Christmas launch for Commodore Max

THE new Commodore Max is to be launched in the UK just before Christmas.

Features will include a 66-key keyboard, complete with four programmable function keys, direct audio and video output and a built-in RF modulator. Twin control ports will allow the use of two joysticks, four paddles or a lightpen.

There is a cartridge slot for games. The Max cartridges will be compatible with the Commodore 64 (reviewed in *Popular Computing Weekly*, September 2).

A mini Basic cartridge option will enable users to write their own programs.

The Commodore Max (formerly called the Vic10 and the Ultimax) has 2K internal memory and a 40 column x 25



Commodore Max.

line display. There are 16 colours, which can be used simultaneously on-screen. Graphics resolution is 320 x 200 pixels.

The Max's sound consists of three independent voices of nine octaves each. There is an envelope generator, a programmable filter and master resonance and volume controls.

The Commodore Max will have a recommended retail price of £110 including VAT.

Panasonic to launch handheld computer in UK

PANASONIC is to launch the RL-H1000 hand held computer in the UK next spring.

The primary unit of the system is based around a 6502 processor with 16K Rom and either 4K or 8K Ram. It also features a built-in 26 character liquid crystal display.

The main advantage of the microcomputer is its size. The primary unit is 9in x 2½in x 3½in and weighs just over 1½lb.

It is powered by five nickel-cadmium rechargeable batteries.

Prestel to give away free adaptors

PRESTEL hopes to give away 100,000 free adaptors, to encourage more home tv owners to use its services. The scheme, which is known as Project Y, will be carried out in collaboration with an unnamed financial institution.

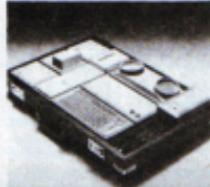
The Prestel adaptors will be offered to the institution's existing customers as part of a package deal.

ies. Mains operation is also possible using an AC adaptor.

The range of peripherals that will be available with the machine includes I/O adaptor, RS232C interface, acoustic modem, tv adaptor and a programmable memory expansion protected by its own back-up battery.

All of these accessories, together with the primary unit, can be fitted inside a specially converted attache case.

The RL-H1000 hand held computer has been on sale in the US since January this year. Current American prices for the primary unit are £225 and £280 for the 4K and 8K versions with peripherals costing between £60 and £200.



Panasonic's handheld micro.

The age of the game

IT is now possible to annihilate space invaders between London and Scotland.

In an experiment being conducted by British Rail, selected trains from Euston to Inverness will carry a Space Invaders machine along with the sandwiches in the buffet car.

The machines have been supplied by Bell Fruit (UK) Ltd and have been installed with the help of area maintenance staff at British Rail's Derby engineering works.

Introduction of the machines was the idea of Peter Dunkerley, a member of BR's Transport Technology Assessment Group. He says "We investigate all kinds of new

technology which may be adapted to increase BR's passenger and freight revenue. The Space Invaders project started following a brainstorming session."

If the scheme is successful the idea may be extended to other long-distance services.



Aliens from Inverness.

Speculation over IBM's Greenock plant

THERE is increasing speculation that IBM's Personal Computer will be produced at its Greenock plant in Scotland.

The IBM Personal Computer, which costs £888, was launched in the US in August 1981. It has yet to be released in the UK, though a limited number of machines have been imported by companies such as KGB Micros.



IBM's Personal Computer launched in the US.

Lynx micro from Computers

THE Lynx, a new micro from Cambridge-based Computers Ltd, will be shown publicly for the first time at the Personal Computer World show in September.

The new micro will cost £150 plus VAT. It includes 48K Ram, an integral power supply and a typewriter-style keyboard. Other features in-

clude high resolution colour graphics and a built-in speaker.

A complete Lynx system, with disc drives and printer, will be on display at the Lynx stand — No 269 — from September 9-12.

More information from Computers Ltd, 36a Hills Road, Cambridge CB2 1LA.

C.P.S. GAMES

Our adventure games and war games run on Alan 400. They consist mainly of graphics. The games are too large for the average 16K computer. We have overcome this problem by running a SERIES OF PROGRAMS IN SUCCESSION. So, although they FIT 16K you get much more.

ADVENTURES

The Ghost of Rasputin: A treasure hunt in a haunted castle.

Baing the Thief: You must steal the tea pot of the Dalai Lama.

The Wizard of Shamb: Travel through jungle and temples to find the elixir of life.

The Fourth Kind: Communicate with extra-terrestrials and voyage through outer space.

The Seven Cities of Cibola: Find the goldfields of the Spanish Jesuits, lost since the 17th century.

The Damned City: Travel through an ants nest where the ants are as big as yourself.

The Tower of Brahma: D&D role playing adventure. Rescue one of your friends from the Khans.

ADVENTURES FOR THE VERY YOUNG

Peter Rabbit and the Magic Carrot.

Peter Rabbit and Father Willow.

Peter Rabbit and the Naughty Owl.

Tummy Digs goes shopping.

Tummy Digs goes walking.

The kids are given a lead and invent the story of the adventure themselves.

WAR GAMES

King Arthur: 6th century war game.

Battle of the Bulge: 1944-1945.

Battle of the River Plate: Well known, full graphics.

Convoy: Full graphics, a battle between a convoy and a wolf pack.

All games at £9.50

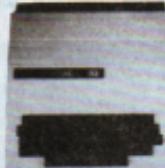
Adventures for the very young £4.50

**C.P.S. 14 Britton Street
London EC1M 5NQ (01-251 3090)**

DRAGONS SIGHTED IN LONDON



BUT DON'T BE AFRAID - THEY'RE
FRIENDLY!



DRAGON 32
THE BRITISH-MADE 32K
HOME COMPUTER FOR
ONLY

£199.50
INC VAT

A breakthrough in home computing, the Dragon 32 has all these features: 32K RAM memory, expandable to 64K. Advanced 6502EC microprocessor; extended memory colour Basic - nine colours; five graphics resolutions (up to 256 x 192 hi-res); professional quality keyboard - uses almost any TV and cassette; sockets for game cartridges, joysticks, cassette, parallel printer; 160 page Basic manual. All for the unbelievably low price of £199.50 inc VAT. If you believe in dragons phone us now.

S. W. WINTER & CO. LTD.
105 WESTMINSTER BRIDGE ROAD, LONDON SE1 01-433 9611

NAME	Please send me	
ADDRESS	Dragon 32 @ £199.50 each	
X	Dragon Joystick @ £19.95 each	
Cassettes		
X	Cassette lead @ £2.50 each	
CARTRIDGES		
X	Street Atlas @ £24.95 each	
MONSTER MINE @ £19.95 each		
MONSTERLAND @ £19.95 each		
Cassettes		
Special selection 1 @ £7.95 each		
Special selection 2 @ £7.95 each		
Special selection 3 @ £7.95 each		
I enclose cheque/PO for.....		
PHONE US FOR A COMPLETE LIST OF GAMES AND SOFTWARE		
Add £3 P&P for each Dragon 32		



Discount Club



To have these top cassettes sent to your home all you have to do is fill in the form below and send it to *Popular Computing Weekly*.

Discount Club Order Form

Quantity	Cassette	Price
	Ship of the Line 48K	£3.30
	Ship of the Line 16K	£3.95
	Fruit Machine/Sub	£3.95
	Cash Control	£8.90
	Monster Mine	£4.25

Total £

I enclose a cheque, made payable to *Popular Computing Weekly Discount Club*, for £.....

Name.....

Address.....

All prices are inclusive of VAT and postage.

Send this order form to *Popular Computing Weekly*, *Discount Club*, Hobhouse Court, 19 Whitcomb Street, London WC2 7HF.

Letters

write to Letters, Popular Computing Weekly, Hobhouse Court, 19 Whitcomb Street, London WC2

Riding the gravy train

I enjoyed Jeremy Ruston's "Voyager" program in the July 22 issue, but the message "You missed — see you in 30 light years" is bad enough to be a quote from Star Wars. As anyone who has ever picked up a physics text book (or watched Star Trek for that matter) could tell you, a light year is a measurement of distance.

Also, why our frozen food should go off because we went fast is beyond me. Might I suggest that the fuel ran out first.

I also enjoyed Reverse by Mike Berry. I hope you can continue the standard.

Peter Greenall
Leicester

Or hitching with the dragon?

It surprised me after reading your review of the Dragon 32 and associated comments and letters from your readers, that no one has spotted one essential fact. The Dragon is not only in direct competition with the Tandy Colour Computer, it also consists of exactly the same technology, with very few exceptions.

The 16K extended Microsoft Basic, with its special commands for 256x192 graphics and one channel sound, is to the byte the same as used on the pre-existing Tandy. The low quality screen display and Rom port are also reminders of that machine.

I owned a Colour Computer for some eleven months and found it to be under-supported and of low quality components ie clickity-clackity keyboard and no parallel port. When I moved here from the US, I sold it at a loss and chose what to me seemed to be a high quality machine with future options for expansion — the BBC model B. Now I see that the ghost of my former computer, albeit with a fairly good keyboard and Centronics port, has followed me over the Atlantic.

PS if the Rom port is the same as the Tandy, and I see no reason why it would not be, then Dragon owners will find there is a ready bank of software at their nearest Tandy dealer.

William Barnes
5 Shaftesbury Villas
Allen Street
London W8

Spectrum bugs wanted

Paul Harrison in this column (August 12) reported a 'bug' in his Spectrum — well, all of the presently available machines have it and more!

Paul noticed that if the 'cursor line' was set to 'one after' the last line of a program then the edit key give an edit-line containing a 'cursor'. His example was to enter:

100 PRINT — ENTER
101 — ENTER
EDIT

and line 100 would be brought down with the 'cursor' (but not when using 255 and 256). So what is happening?

In the above case, the 'cursor line' is 100 when the first Enter is keyed. It becomes 101 when the second Enter is keyed but when Edit is used there is no line 101 and the last line of the program, ie 100, is called down.

Now for the problem. The programmer of the Rom decided that Edit lines shall not have a 'cursor'. He says that before printing an Edit line the system variable E-PPC, that holds the cursor line number, will be decremented — but he forgot that the last line of the program might have a line number that is 'E-PPC — 1'. Hence on those special occasions a 'cursor' is printed.

The solution is probably quite simple — change the instruction line at OFDE & from Call 1855 to Call 1860.

And for two further bugs. Try replying to 'scroll?' with 'caps lock', 'graphics', or 'shift & symbol shift'. These results are interesting and it is most surprising that the Spectrum does not crash.

The other bug is to ponder over why 'Int - 65536 = -1'. No doubt other 'bugs' will

appear and I would very much like to hear about them.

Ian Logan
24 Nurses Lane
Skellingthorpe
Lincoln LN6 5TT

Time gentlemen —please

My Spectrum has been on Morder from Sinclair for eight weeks now. Having contacted them recently, I am informed that rumours of the backlog being eliminated are "greatly exaggerated". How much longer must I wait, I wonder.

When my Spectrum does arrive, however, I would like to know what amendments I will need to make to the numerous programs I have collected for the ZX81 so that they will run on the Spectrum. I have not seen this aspect of upgrading mentioned or dealt with in any magazine. Can you help?

B Davies
2 Juniper Crescent
Apsley Park
Longthorpe
Peterborough PE3 0Y

Most ZX81 Basic programs will run on the Spectrum with little or no modification. But, you will have to add your own commands to take advantage of the Spectrum's colour and sound facilities.

The simplest solution may be to use East London Robotics' Slowloader, which allows ZX81 software to run on the Spectrum. It costs about £10 and is available from East London Robotics, Finlandia House, 14 Darwell Close, East Ham, London E6.

Cut-outs — the moaning

I am writing to make some constructive criticism, not just complain.

I have found myself in agreement with your editors and they have provided inspiration on many occasions. However, your editorial of July 29 exasperated me.

Motivated by a previous editorial extolling the virtues of machine code, I attempted

to develop my limited skills by writing a machine code program of my own. But, before I could send it off in all its glory for your hoped for approval, I read "Moonlander, Space Invader and Fruit Machine programs abound". My program was a version of Fruit Machine. Oh Well!

Having my work accepted is a great boost to my enthusiasm. Please encourage us, but be tolerant when we use an "oldie" idea to try and improve our skills. Soon, I hope, both my skills and my ideas will be increased.

Second moan. Could you print cut-out forms and competition entries on the back of advertising pages. I do not want to enter the "Spectrum" competition as it would mean losing large sections of articles that I wish to keep for future reference. Likewise your Reader Survey.

When you began, you ran competitions on a theme. Could you not repeat this, perhaps on a monthly basis? Then you could print two or three of the best entries in future issues. This would give us some motivation, even if the only "prize" was possible publication.

Ray Elder
1 Periton Court
Parkhouse Road
Minehead
Somerset

We do try and print cut-out forms on the back of advertising pages, but the exigencies of production are such that it is not always possible. As for theme competitions, please write in and let us know your views.

On-screen Images

After typing in the short machine code routine in the Instant screen article in July 1, I noticed that changing 3E in byte 16526 to C6 produced an inverse image of the screen. All the information in table 1 still applies to the routine.

Patrick Budden
Wolfeton
Dorchester DT2 9QN

COVER STORY

Pteragon

A new game for BBC micro by A Snell

The sudden insistent beep of the search radar alarm breaks the silence in the cockpit of the solar-powered skimmer. As a newcomer to the Planetary Defence Force, you tense to full alertness — this is your first mission.

The recently established colony on the planet Beta Strigidae has been under attack from huge winged reptiles — the semi-sentient Pteragon. Your task is to destroy the incoming waves of monsters before they reach the settlements. Each member of the PDF must deal with five of the creatures in each attack, and there is intense competition within the force to shoot them down in as short a time as possible.

The sound of powerful wingbeats can be heard as the first of the Pteragon comes into view through your craft's forward ports, you must manoeuvre quickly to frame the monster in your sights before it flies past. The 'A' and 'Z' keys move the skimmer up and down, and the '>' and '<' keys move you right and left. The space bar fires your energy beam weapon, and if you are a good shot will vapourise the Pteragon.

If the Pteragon moves out of your field of view for too long, you will lose it and incur a time penalty. The score shown on your display is the time taken to shoot down the number of Pteragon shown in the HITS column, plus any time penalties for missed monsters. RECORD gives the fastest time achieved so far — the time that you must beat.

This program has been especially written for the BBC micro, and makes use of many of the advanced sound and graphics features of this machine. It will run on a model B, or on a model A with 32K of memory.

A look at the program listing will show that it has been written almost entirely as a set of procedures, and hence should be very easy to follow. Of special interest is the simple procedure to draw the Pteragon, at line 2000. This makes use of the PLOT 86 statement, which plots using the logical inverse colour. It enables the creature to move behind the foreground features on the screen by an appropriate choice of colour definitions in the initialisation procedure at line 5000.

If a longer game is required, the number of Pteragon to be shot down can be increased by changing the limit in line 50. The procedure called 'rules' at line 4500 has been left out of this listing to keep the length down, but can easily be written in to give a brief description of the operation of the controls before the game starts.

This program has also been successfully run using joysticks on the model B. Lines 550-570 are replaced by:

```
559 F=ADVAL(9)  
560 IF F=1 THEN PROCchase  
579 IF TIME>T1+150 THEN PROCchase  
and the procedure at line 4000 becomes:  
4000 DEF PROCchase  
4010 X=X+36-INT(ADVAL(2)/999)  
4020 Y=Y+36-INT(ADVAL(2)/999)  
4030 ENDPROC
```



Our classifieds are faster.

Do you want to sell your computer and buy a bigger and better one?

Have you ever thought of trying to make some money out of selling tapes of your own programs?

Whatever it is you want to buy or sell why not use our classified pages?

It has to be better than waiting for up to nine weeks to get into one of the old monthly magazines.

Not only that, but our rates are very reasonable.

For private individuals it only costs 20p per word, with a minimum of 10 words.

We can make it so cheap because we charge companies using the classified columns 40p per word.

The classified pages can be used for semi-display advertising.

The cost for this is £10 per single column centimetre, with a minimum charge of £30.

All copy for the classified pages must be pre-paid. (You'll find a handy form on page 25.)

Cheques and postal orders should be made out to *Popular Computing Weekly*. Your advertisement should arrive at least two weeks before the publication date.

If you have any queries regarding Classified or semi-display advertising please call Alastair Macintosh on 01-930 3840

Popular Computing Weekly.
The fast one.

Street Life

Oversize aquatic horror fitted with remote control

David Kelly talks to *Ulvatech* about a rival to *Jaws*.

If only Stephen Spielberg had known what he was starting when he filmed *Jaws*.

Ever since the killer fish made its debut, the screens have been filled with a succession of sharks, whales, squids, piranhas and even a deadly beach ('just when you thought it was safe to go back in the water, you can't even get across the beach').

Now *The Pike* is planned — featuring a 12 ft-long micro-controlled monster. Providing the £2.5m budget can be raised, the film should go into production later this year, for release in 1983.

Work on the giant fish, controlled by an Acorn Atom, has been completed and the creature is undergoing 'sea' trials.

The huge man-eating pike has been designed and built by *Ulvatech* Ltd. Based in Ulverston, Cumbria, the four-year old company was formed by several former employees of Vickers Oceanic.

George Colquhoun, the company's managing director explained that *Ulvatech* specialises in underwater remote control devices and sonar equipment. Its most successful product is a scanning profiler which produces sea-bed cross-sections using a device incorporating a 6502 processor on the sea floor coupled to an LSI 11 surface computer.

It was on the strength of this device that

the Manchester-based film company, City Major, commissioned *Ulvatech* to build the giant pike.

Building the over-size aquatic horror was not easy. The basic design was produced by *Ulvatech* but the body of the fish was sculpted by Manchester artist Charles Wyatt.

To aid in the design, two live captive pike were studied and their movements video-taped to discover the secrets of their swimming action.

From the outside, the *Ulvatech* pike is indistinguishable from the real thing only by its size.

The pike has been designed to swim under its own power. The tail and body of the 600lb model bend back and fore to propel it through the water. The motive force comes from a pneumatic system powered by compressed air containers inside the fish.

Christopher Edwards, *Ulvatech*'s electronics expert, explained how the Acorn Atom is used to control the pike.

"We fitted one of Acorn's versatile interface boards to the Atom and used the 20mA serial output.

"The micro regularly sends an eight-line data bit to each address on the pike. The tail, body, rear fins, front fins, jaw and eyes can all be controlled."

There are 16 possible positions for each part of the fish. The control information is passed from the computer to the fish

through a thin umbilical wire. Of the eight data lines, four are used to show the address and four are used to set the position of that address.

The feedback output from each part of the pike is converted from analogue to digital. It is continuously matched in a central asynchronous decoder with the corresponding input from the micro.

Thus the positions of each part of the fish are updated eight times each second using information from the microcomputer.

Before the pike can be made to swim, the Atom generates a data table. Amplitude values are then picked off this table, held in Ram by the computer, and sent to the fish to produce the swimming movements.

"Obviously" said Christopher "it is a fairly unsophisticated data bank but, since each part of the fish can only register 16 positions, a more accurate system was unnecessary.

"We started the electronics part of the pike in January. The whole fish was completed and tested in our tank in May."

Since then the pike has undergone successful trials on Lake Windermere. According to Christopher Edwards the fish cost about £30,000 to make.

He says "It was great to work on. If we were all rich we could afford to play with it to our heart's content."

The giant pike is currently residing in *Ulvatech*'s warehouse, waiting for City Major to begin filming.

Errata

Jonathan Briggs is not forming a North-west London Spectrum User Group (*Popular Computing Weekly*, August 19). Please do not attempt to contact him.

Joan Collins (right), examining the giant Pike (left), whose exterior skin was constructed using a special tear-resistant silicon rubber reinforced with cotton muslin.



Copyright Lakeland Photographic, Kendal, Cumbria.

Jupiter Ace makes Forth bid for stardom

Boris Allan presents the first review of the new Jupiter Ace and explains its use of the language Forth.

The Jupiter Ace is a microcomputer from two leading designers of the Sinclair Spectrum (Richard Altwasser and Steve Vickers). The Ace is based on the Z80A microprocessor, with 8K bytes of Rom and 3K bytes of Ram; has a Spectrum-style keyboard (though it is promised to be more positive); has a proper memory-mapped screen (unlike the Spectrum), with 32 x 24 display on an ordinary television; and has sound and cassette facilities similar to those of the Spectrum, but no colour. It costs £89.95.

If the Ace sounds like a Spectrum under a different cover, it is not: the Ace is set apart from the Spectrum, and all other small microcomputers, by the use of the programming language Forth alone.

To emphasise, you cannot program the Ace in Basic, you have to use this rather different language Forth. Though it is an uncommon language, many people have recently become interested in Forth, and versions have appeared inter alia for the Atom, ZX81, MZ80K, BBC computer, and there is Forth available on the latest version of the Osborne 1.

For Forth to appear on the Osborne (a business machine) implies that this is a language which is beginning to make commercial sense.

When confronted with a machine with higher resolution graphics, and a claim from its designers that "Ace is ideal for games", the question is "why Forth and not Basic? Is Forth an easy language for beginners?"

In a way some people are afraid of Forth — a fear of the unknown? — as it seems

unnatural to type in

2 5 +

when you really mean

2 + 5

If I ask you to add 2 and 5, you answer 7; if I ask you to take 2 and 5 and add them together, you will still answer 7. When you add two numbers you are doing something to those two numbers to produce a result (another number). In Forth it is clear you are doing something to numbers, to produce another number.

Forth says here are two numbers, add them together; 2 and 5 are the numbers, and + means "add them together"

2 5 +

It's not all that strange really.

What is Forth? In essence Forth is a computer language which is truly interactive (almost as if it were a Basic in instant mode all the time), and it is a language which is truly extensible.

In Basic it is possible to add new functions and procedures, but very difficult to add new commands unless one uses machine code routines.

Forth is a very efficient language, much speedier than most Basics. The 8K of Rom in the Ace corresponds to a far greater amount of Rom for any Basic, and the 3K of Ram is worth more than 3K of Ram if

Reviews



used by a Basic program.

The version of Forth used by the Ace is based, loosely, on two standards, FIG-Forth (Forth Interest Group) and Forth-79, but with important modifications. All the modifications seem to be improvements. I will examine some of them in more detail after I have looked at more general features, common to most Forths.

Forth is most notable for the way in which it uses basic functions (or procedures) to produce more powerful functions, so that, finally, the whole program (or 'application') is just one all-embracing function (or 'word').

Forth also uses a stack, and the user can control what happens on the stack. Think of the stack as a pile of numbers, on which you can put more numbers or take numbers off, but at one end only.

Suppose I want to multiply two numbers together. In Forth I say "here are two numbers, multiply them together"

2 5 *

and this line means take the numbers 2 and 5, multiply them together, and finally print out the result (which is what the '*' means).

Forth copes with this line by starting at the left, and moving right. The first item encountered is "2" and this is recognised as a number. The number 2 is "stacked" for later use. The second item is "5". This is a number and is stacked on top of the 2, for later use. When the program meets the word "*" it knows that this means multiply. It multiplies the two numbers on the top of the stack (ie 2 and 5), so removes them and places the answer on top. When the word "*" is met it is understood to mean print the value of the number on top of the stack and remove the number.

The word "*" is not very self-explanatory, but in Forth if you do not like the name of a function/word you can change it, eg:

: PRINT . CR ;

and now the word "Print" means do "*" and then carriage return ("Cr"). The sum now looks like

2 5 PRINT

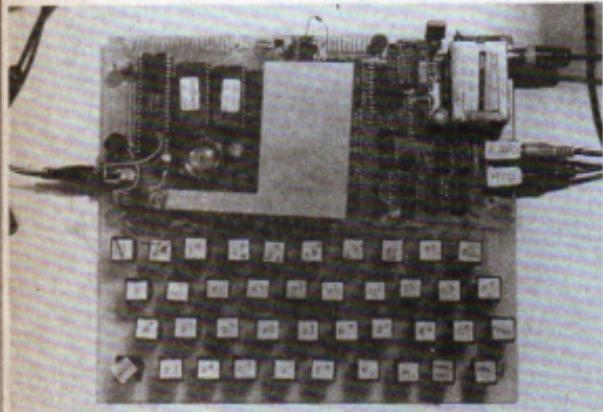
and it is possible to continue.

It would be nice to find the square of the number 7, and it could be done by

7 7 * PRINT

or, possibly

7 DUP * PRINT



Interior layout of a pre-production model of the Jupiter Ace.

where the word "Dup" means take the number on top of the stack (in this case 7) and put another copy on top of it (Duplicate). It would be nice to do this by just saying

7 SQUARED

and one can by

: SQUARED DUP * PRINT;
which is a 'colon' of the word "Squared", which definition uses a word "PRINT" defined earlier. And so the process can continue.

In normal Forths editing words and

Vickers version of Forth (eg Redefine or Fast or Edit) which are worth incorporating in other versions.

I have found that Forth on microcomputers is not (generally speaking) user friendly — on most systems it is only too easy to find oneself in a situation where the only thing one can do is switch off and reload the Forth system. The Ace system is user-friendly, and difficult to crash.

The question of how user-friendly is the Ace is important because, being aimed, as it is, at the cheap end of the market and

aries and applications, often this will not be necessary. As Forth itself is in ROM, if the system does crash — Steve Vickers and I had great difficulty in performing this feat — if it does crash, Forth is not lost.

For a beginner, I do not think that Forth is intrinsically any more difficult to learn than Basic, if it is approached in the right manner; but even if it is not more difficult, is it any better? Altwasser and Vickers are, I believe, sincere in their belief that Forth is intrinsically a better language, and in one sense I agree. Forth is better because as we have seen it is possible (once the groundwork has been done) to perform complicated operations with a minimum of programming. Some of the queries come with "once the groundwork has been done".

The word "Squared" used another word "Print" and another word might use them both, but "Squared" could not use "Print" until we had defined "Print"; though with Ace Forth it is possible to give a dummy definition to "Print" (eg : PRINT :) and then later use "Redefine" to give a proper definition. A possible problem with this approach (for novice and not so novice) is the "design at the keyboard" syndrome. That is, it is easy to get tied up in the minutiae and lose sight of the overall structure. This is also a problem with most other languages, whether they be called structured or not.

I have a feeling that the Ace might well find a market as a cheap way of learning Forth (schools, colleges, and businessmen?) for some, and for others a cheap way of obtaining a machine with the speed of Forth (not far slower than poorly written machine code) for control applications. For those who would like to use the Ace for control applications, the way in which the Ace talks to the world is obviously important. The Ace talks via the television, and a rear port — the one I saw connected to the Ram pack. Richard Altwasser told me that they planned to provide means of coupling up to all the main interfaces one would need for use in control applications.



Steve Vickers puts the Jupiter Ace's keyboard through its paces.

applications can be tedious (for reasons too tedious to explain) and the designers of the Ace had come to the same conclusion. In normal Forths it is possible to keep a copy of a definition on what is termed a 'screen', but once the definition has been compiled the internal workings cannot be changed. With the Ace there is no need to keep a separate copy of the listing of the definition — in fact one does not, in any case — and one can edit, list, and "Redefine" words (replace an earlier version of a word by the latest definition, and redefine any other definitions that are necessary).

Forths do not normally have floating point operations as part of the standard set of words, but the Ace has five special floating point words for use on six significant digits (1E64 to 1E-64). Ace Forth also has inbuilt array using features.

In many respects the Ace has many of the good features of the Spectrum, very good cassette saving, loading, and verifying, good line editing, good user defined graphics capabilities. As a way of using Forth, it is excellent. Apart from these resemblances, the Ace is a very different machine because it is in black and white (though a colour facility is being considered), and it can use the ZX81 16K Ram pack (plus many of the other ZX81 add-ons). An easy machine to use.

Though the machine I used was not the final production version I have no doubts about the eventual production machine. I am sure that as a Forth machine it will be excellent. I think that given my use of Forth on the Ace there are many ideas in Steve

competing with cheaper black and white computers, and slightly more expensive colour computers, and as it uses what is a novel language, the Ace has to be user-friendly to succeed. Once one has the dictionary of words — tailored to one's own interest — to create sophisticated applications should not be as difficult as creating large programs in Basic. One word can stand for a program or a standard (often complex) facility.

Though it is possible to merge dictio-



Richard Altwasser "providing the Ace with interfaces for use in control applications".

Open Forum

Open Forum is for you to publish your programs and ideas.

It is important that your programs are bug free before you send them in. We cannot test all of them.

Contributions should be sent to: Popular Computing Weekly, Hobhouse Court,
19 Whitcomb Street, London WC2H 7HF.

How to contribute

Each week the editor goes through all the programs that you send to Open Forum in order to find the Program of the Week.

The author of that program will qualify for DOUBLE the usual fee we pay for published programs. (The usual fee is £10.)

Presentation hints

Programs which are most likely to be considered for the Program of the Week will be computer printed and accompanied by a cassette.

The program will be well documented, the documentation being typed with a double spacing between each line.

The documentation should start with a general description of the program and then give some detail of how the program has been constructed and of its special features.

Listings taken from a ZX Printer should be cut into convenient lengths and carefully stuck down on to white paper, avoiding any creasing.

Please enclose a stamped, self-addressed envelope.

Roman Numerals Converter

on ZX81

Roman numerals are found in the flyleaves of old books and on many old films, as well as on stones of old buildings and so on. In these contexts, they usually represent dates.

These strings of letters are often confusing and difficult to interpret. The following program interprets them for you. In addition, it will also convert any decimal number from 1 through 4999 to Roman numerals.

Program notes

Lines

```
00 LET RS="XCMVLD"
110 LET XS=""
120 INPUT AS
130 IF AS="STOP" THEN STOP
140 SCROLL
150 PRINT AS,
```

```
160 IF AS>"A" THEN GOTO 350
170 FOR J=1 TO LEN AS
180 LET YS=""
190 LET A=VAL AS(LEN AS-J+1)
200 IF A AND A<5 OR A=9 THEN LET
    YS=YS+RS(J)
210 IF A=3 THEN GOTO 280
220 LET A=A-1
230 IF A>4 THEN GOTO 200
240 LET XS=YS+XS
250 NEXT J
260 PRINT XS
270 GOTO 100
280 IF A=9 THEN GOTO 330
290 IF J=4 THEN GOTO 220
300 LET YS=YS+RS(J-4)
310 LET A=A-5
320 GOTO 230
330 LET YS=YS+RS(J-1)
340 GOTO 240
350 IF AS<"C" THEN GOTO 100
360 LET I=1
370 LET V=5
380 LET X=10
390 LET L=50
400 LET C=100
410 LET D=500
420 LET M=1000
430 LET T=0
440 FOR J=2 TO LEN AS
450 LET B=VAL AS(J-1)
460 LET E=VAL AS(J)
470 IF E>B THEN GOTO 530
480 LET T=T+B
490 NEXT J
500 LET T=VAL AS(LEN AS)+T
510 PRINT T
520 GOTO 100
530 FOR J=1 TO 1 STEP -1
540 IF AS(J)> AS(J-1), THEN GOTO 480
550 LET E=VAL AS(J)
560 LET T=T-2*E
570 NEXT K
580 GOTO 400
```

To operate this program, Run, and enter either:

- i) a decimal number (not exceeding 4999) for conversion to Roman numerals, or
- ii) a Roman numeral (not exceeding MMMMMCMXCIX) for conversion to decimal.

Press newline. The number you entered is displayed on the left, the conversion on the right.

ZX Artist

on ZX81

This art program gives the user all the usual things other art programs give plus the facility to draw a straight line from one

given point to another, draw a circle and provide the co-ordinates for plotting.

The program also incorporates one point plotting, the ability to draw lines using the cursor keys, copy pictures on to the printer and store pictures on tape.

When the program is Run the user is given the choice of 10 options to choose from.

Option "0" will first ask the user for a starting point (X and Y co-ordinates) and then hand all control over to the user with the cursor keys.

Option selection

The user can leave this part of the program any time and return to the option section of the program by pressing the "9" key.

Option "1" will ask for co-ordinates and plot a single pixel on that point.

Option "2" will ask for co-ordinates and unplot a single pixel on that point.

Option "3" will ask for starting co-ordinates and finishing co-ordinates and will plot a line between them.

Option "4" will do the same as option "3" but will unplot a line between the co-ordinates.

Option "5" will ask for X and Y co-ordinates and a radius. It will then ask for how much of a circle the user wants by using the numbers 0-40, 0-10 will plot the first quarter of a circle, 10-20 the second etc. 0-20 would plot the first half of the circle.

Option "6" does the same as option "5" but unplots the circle.

Option "7" copies the border and its contents on to the printer.

Option "8" saves the program and the screen on to tape.

Option "9" stops the program.

Program notes

Lines

```
18-198 set up the screen.
119-198 sort out the options.
200-278 provide option "0".
1980-1981 provide option "1".
2000-2010 provide option "2".
3880-3210 provide option "3".
4380-4310 provide option "4".
5880-5160 provide option "5".
5880-6010 provide option "6".
7880-7170 provide option "7".
8300-8340 save the program and screen.
9300 stops the program.
```


Open Forum

each note and on line 150, 12 is subtracted to lower the pitch one octave. On line 340, 24 is subtracted to lower the pitch by two octaves.

The N loops are equal to the number of chords in the Data lines. The K loops select the entire six notes in each chord (see line 150), and only the first three notes (line 340).

When the program is Run you should hear a fair rendition of House of the Rising Sun, followed by a short pause for applause. Next, something which sounds like Wimoweh and goes on for ever (just like the real thing). Hold down any key for a few seconds and the program jumps to the random composer routine.

The last short program is purely visual and produces some very effective moving displays. Run it and watch for a while then experiment with new values for the variables on line 100.

The values for e and t should always be equal to or greater than the x,y values.

Cards

on BBC Micro

The enclosed code uses a very efficient 'sample without replacement' technique to simulate dealing a deck of cards.

Call *Procmakedeck* initially and whenever you want a fresh deck. Use *Findat* to return the number of the next card dealt. At any time, *Deck(0)* holds the number of cards left in the pack, N say, and *Deck(N+1)*, *Deck(N+2)* etc is a complete history of previously-dealt cards.

The algorithm can easily be adapted for other purposes such as pools coupons — simply set *Ncard* to 55 or whatever value is required and redimension *Deck* accordingly.

Casio

on BBC Micro

With the advent of user defined graphics on most of the new micros, there have been a number of sophisticated programs to decode binary patterns into numbers which are then used to define the shape.

These programs can be useful. However, I know that there are thousands of people who want to define a shape simply while in the middle of writing a program.

I use either a Casio FX602P calculator with a simple binary to decimal routine or I program a function key on my BBC micro with either of these methods.

If you know the method, it is easy to write a program to convert decimal to binary and back. However, it is not quite so easy to do it on one line (as is needed for a function key). The reason for this is that *If Then* statements are very difficult to implement (since if the outcome is false, there is no new line number to execute, so the routine terminates) and the *Else* command can over-complicate things.

Line 40 overcomes this problem, as the loop selects the binary digits one at a time

```

10 REM Auto Composer
11 REM by Ron Smith
12 REM assign chords a number
13 REM 0=1, 1=2, 2=3, 3=4, 4=5, 5=6
14 REM LET D=0
15 REM LET G=1
16 REM LET C=2
17 REM LET B=3
18 REM LET A=4
19 REM LET E=5
20 REM LET F=6
21 REM DIM C(6) 121
22 REM DIM G(6) 121
23 REM DIM D(6) 121
24 REM DIM B(6) 121
25 REM DIM A(6) 121
26 REM DIM E(6) 121
27 REM DIM F(6) 121
28 REM C(0) = "121212121212121212"
29 REM C(1) = "121212121212121212"
30 REM C(2) = "121212121212121212"
31 REM C(3) = "121212121212121212"
32 REM C(4) = "121212121212121212"
33 REM C(5) = "121212121212121212"
34 REM C(6) = "121212121212121212"
35 REM C(7) = "121212121212121212"
36 REM C(8) = "121212121212121212"
37 REM C(9) = "121212121212121212"
38 REM C(10) = "121212121212121212"
39 REM C(11) = "121212121212121212"
40 REM C(12) = "121212121212121212"
41 REM C(13) = "121212121212121212"
42 REM C(14) = "121212121212121212"
43 REM C(15) = "121212121212121212"
44 REM C(16) = "121212121212121212"
45 REM C(17) = "121212121212121212"
46 REM C(18) = "121212121212121212"
47 REM C(19) = "121212121212121212"
48 REM C(20) = "121212121212121212"
49 REM C(21) = "121212121212121212"
50 REM C(22) = "121212121212121212"
51 REM C(23) = "121212121212121212"
52 REM C(24) = "121212121212121212"
53 REM C(25) = "121212121212121212"
54 REM C(26) = "121212121212121212"
55 REM C(27) = "121212121212121212"
56 REM C(28) = "121212121212121212"
57 REM C(29) = "121212121212121212"
58 REM C(30) = "121212121212121212"
59 REM C(31) = "121212121212121212"
60 REM C(32) = "121212121212121212"
61 REM C(33) = "121212121212121212"
62 REM C(34) = "121212121212121212"
63 REM C(35) = "121212121212121212"
64 REM C(36) = "121212121212121212"
65 REM C(37) = "121212121212121212"
66 REM C(38) = "121212121212121212"
67 REM C(39) = "121212121212121212"
68 REM C(40) = "121212121212121212"
69 REM C(41) = "121212121212121212"
70 REM C(42) = "121212121212121212"
71 REM C(43) = "121212121212121212"
72 REM C(44) = "121212121212121212"
73 REM C(45) = "121212121212121212"
74 REM C(46) = "121212121212121212"
75 REM C(47) = "121212121212121212"
76 REM C(48) = "121212121212121212"
77 REM C(49) = "121212121212121212"
78 REM C(50) = "121212121212121212"
79 REM C(51) = "121212121212121212"
80 REM C(52) = "121212121212121212"
81 REM C(53) = "121212121212121212"
82 REM C(54) = "121212121212121212"
83 REM C(55) = "121212121212121212"
84 REM C(56) = "121212121212121212"
85 REM C(57) = "121212121212121212"
86 REM C(58) = "121212121212121212"
87 REM C(59) = "121212121212121212"
88 REM C(60) = "121212121212121212"
89 REM C(61) = "121212121212121212"
90 REM C(62) = "121212121212121212"
91 REM C(63) = "121212121212121212"
92 REM C(64) = "121212121212121212"
93 REM C(65) = "121212121212121212"
94 REM C(66) = "121212121212121212"
95 REM C(67) = "121212121212121212"
96 REM C(68) = "121212121212121212"
97 REM C(69) = "121212121212121212"
98 REM C(70) = "121212121212121212"
99 REM C(71) = "121212121212121212"
100 REM C(72) = "121212121212121212"
101 REM C(73) = "121212121212121212"
102 REM C(74) = "121212121212121212"
103 REM C(75) = "121212121212121212"
104 REM C(76) = "121212121212121212"
105 REM C(77) = "121212121212121212"
106 REM C(78) = "121212121212121212"
107 REM C(79) = "121212121212121212"
108 REM C(80) = "121212121212121212"
109 REM C(81) = "121212121212121212"
110 REM C(82) = "121212121212121212"
111 REM C(83) = "121212121212121212"
112 REM C(84) = "121212121212121212"
113 REM C(85) = "121212121212121212"
114 REM C(86) = "121212121212121212"
115 REM C(87) = "121212121212121212"
116 REM C(88) = "121212121212121212"
117 REM C(89) = "121212121212121212"
118 REM C(90) = "121212121212121212"
119 REM C(91) = "121212121212121212"
120 REM C(92) = "121212121212121212"
121 REM C(93) = "121212121212121212"
122 REM C(94) = "121212121212121212"
123 REM C(95) = "121212121212121212"
124 REM C(96) = "121212121212121212"
125 REM C(97) = "121212121212121212"
126 REM C(98) = "121212121212121212"
127 REM C(99) = "121212121212121212"
128 REM C(100) = "121212121212121212"
129 REM C(101) = "121212121212121212"
130 REM C(102) = "121212121212121212"
131 REM C(103) = "121212121212121212"
132 REM C(104) = "121212121212121212"
133 REM C(105) = "121212121212121212"
134 REM C(106) = "121212121212121212"
135 REM C(107) = "121212121212121212"
136 REM C(108) = "121212121212121212"
137 REM C(109) = "121212121212121212"
138 REM C(110) = "121212121212121212"
139 REM C(111) = "121212121212121212"
140 REM C(112) = "121212121212121212"
141 REM C(113) = "121212121212121212"
142 REM C(114) = "121212121212121212"
143 REM C(115) = "121212121212121212"
144 REM C(116) = "121212121212121212"
145 REM C(117) = "121212121212121212"
146 REM C(118) = "121212121212121212"
147 REM C(119) = "121212121212121212"
148 REM C(120) = "121212121212121212"
149 REM C(121) = "121212121212121212"
150 REM C(122) = "121212121212121212"
151 REM C(123) = "121212121212121212"
152 REM C(124) = "121212121212121212"
153 REM C(125) = "121212121212121212"
154 REM C(126) = "121212121212121212"
155 REM C(127) = "121212121212121212"
156 REM C(128) = "121212121212121212"
157 REM C(129) = "121212121212121212"
158 REM C(130) = "121212121212121212"
159 REM C(131) = "121212121212121212"
160 REM C(132) = "121212121212121212"
161 REM C(133) = "121212121212121212"
162 REM C(134) = "121212121212121212"
163 REM C(135) = "121212121212121212"
164 REM C(136) = "121212121212121212"
165 REM C(137) = "121212121212121212"
166 REM C(138) = "121212121212121212"
167 REM C(139) = "121212121212121212"
168 REM C(140) = "121212121212121212"
169 REM C(141) = "121212121212121212"
170 REM C(142) = "121212121212121212"
171 REM C(143) = "121212121212121212"
172 REM C(144) = "121212121212121212"
173 REM C(145) = "121212121212121212"
174 REM C(146) = "121212121212121212"
175 REM C(147) = "121212121212121212"
176 REM C(148) = "121212121212121212"
177 REM C(149) = "121212121212121212"
178 REM C(150) = "121212121212121212"
179 REM C(151) = "121212121212121212"
180 REM C(152) = "121212121212121212"
181 REM C(153) = "121212121212121212"
182 REM C(154) = "121212121212121212"
183 REM C(155) = "121212121212121212"
184 REM C(156) = "121212121212121212"
185 REM C(157) = "121212121212121212"
186 REM C(158) = "121212121212121212"
187 REM C(159) = "121212121212121212"
188 REM C(160) = "121212121212121212"
189 REM C(161) = "121212121212121212"
190 REM C(162) = "121212121212121212"
191 REM C(163) = "121212121212121212"
192 REM C(164) = "121212121212121212"
193 REM C(165) = "121212121212121212"
194 REM C(166) = "121212121212121212"
195 REM C(167) = "121212121212121212"
196 REM C(168) = "121212121212121212"
197 REM C(169) = "121212121212121212"
198 REM C(170) = "121212121212121212"
199 REM C(171) = "121212121212121212"
200 REM C(172) = "121212121212121212"
201 REM C(173) = "121212121212121212"
202 REM C(174) = "121212121212121212"
203 REM C(175) = "121212121212121212"
204 REM C(176) = "121212121212121212"
205 REM C(177) = "121212121212121212"
206 REM C(178) = "121212121212121212"
207 REM C(179) = "121212121212121212"
208 REM C(180) = "121212121212121212"
209 REM C(181) = "121212121212121212"
210 REM C(182) = "121212121212121212"
211 REM C(183) = "121212121212121212"
212 REM C(184) = "121212121212121212"
213 REM C(185) = "121212121212121212"
214 REM C(186) = "121212121212121212"
215 REM C(187) = "121212121212121212"
216 REM C(188) = "121212121212121212"
217 REM C(189) = "121212121212121212"
218 REM C(190) = "121212121212121212"
219 REM C(191) = "121212121212121212"
220 REM C(192) = "121212121212121212"
221 REM C(193) = "121212121212121212"
222 REM C(194) = "121212121212121212"
223 REM C(195) = "121212121212121212"
224 REM C(196) = "121212121212121212"
225 REM C(197) = "121212121212121212"
226 REM C(198) = "121212121212121212"
227 REM C(199) = "121212121212121212"
228 REM C(200) = "121212121212121212"
229 REM C(201) = "121212121212121212"
230 REM C(202) = "121212121212121212"
231 REM C(203) = "121212121212121212"
232 REM C(204) = "121212121212121212"
233 REM C(205) = "121212121212121212"
234 REM C(206) = "121212121212121212"
235 REM C(207) = "121212121212121212"
236 REM C(208) = "121212121212121212"
237 REM C(209) = "121212121212121212"
238 REM C(210) = "121212121212121212"
239 REM C(211) = "121212121212121212"
240 REM C(212) = "121212121212121212"
241 REM C(213) = "121212121212121212"
242 REM C(214) = "121212121212121212"
243 REM C(215) = "121212121212121212"
244 REM C(216) = "121212121212121212"
245 REM C(217) = "121212121212121212"
246 REM C(218) = "121212121212121212"
247 REM C(219) = "121212121212121212"
248 REM C(220) = "121212121212121212"
249 REM C(221) = "121212121212121212"
250 REM C(222) = "121212121212121212"
251 REM C(223) = "121212121212121212"
252 REM C(224) = "121212121212121212"
253 REM C(225) = "121212121212121212"
254 REM C(226) = "121212121212121212"
255 REM C(227) = "121212121212121212"
256 REM C(228) = "121212121212121212"
257 REM C(229) = "121212121212121212"
258 REM C(230) = "121212121212121212"
259 REM C(231) = "121212121212121212"
260 REM C(232) = "121212121212121212"
261 REM C(233) = "121212121212121212"
262 REM C(234) = "121212121212121212"
263 REM C(235) = "121212121212121212"
264 REM C(236) = "121212121212121212"
265 REM C(237) = "121212121212121212"
266 REM C(238) = "121212121212121212"
267 REM C(239) = "121212121212121212"
268 REM C(240) = "121212121212121212"
269 REM C(241) = "121212121212121212"
270 REM C(242) = "121212121212121212"
271 REM C(243) = "121212121212121212"
272 REM C(244) = "121212121212121212"
273 REM C(245) = "121212121212121212"
274 REM C(246) = "121212121212121212"
275 REM C(247) = "121212121212121212"
276 REM C(248) = "121212121212121212"
277 REM C(249) = "121212121212121212"
278 REM C(250) = "121212121212121212"
279 REM C(251) = "121212121212121212"
280 REM C(252) = "121212121212121212"
281 REM C(253) = "121212121212121212"
282 REM C(254) = "121212121212121212"
283 REM C(255) = "121212121212121212"
284 REM C(256) = "121212121212121212"
285 REM C(257) = "121212121212121212"
286 REM C(258) = "121212121212121212"
287 REM C(259) = "121212121212121212"
288 REM C(260) = "121212121212121212"
289 REM C(261) = "121212121212121212"
290 REM C(262) = "121212121212121212"
291 REM C(263) = "121212121212121212"
292 REM C(264) = "121212121212121212"
293 REM C(265) = "121212121212121212"
294 REM C(266) = "121212121212121212"
295 REM C(267) = "121212121212121212"
296 REM C(268) = "121212121212121212"
297 REM C(269) = "121212121212121212"
298 REM C(270) = "121212121212121212"
299 REM C(271) = "121212121212121212"
200 REM C(272) = "121212121212121212"
201 REM C(273) = "121212121212121212"
202 REM C(274) = "121212121212121212"
203 REM C(275) = "121212121212121212"
204 REM C(276) = "121212121212121212"
205 REM C(277) = "121212121212121212"
206 REM C(278) = "121212121212121212"
207 REM C(279) = "121212121212121212"
208 REM C(280) = "121212121212121212"
209 REM C(281) = "121212121212121212"
210 REM C(282) = "121212121212121212"
211 REM C(283) = "121212121212121212"
212 REM C(284) = "121212121212121212"
213 REM C(285) = "121212121212121212"
214 REM C(286) = "121212121212121212"
215 REM C(287) = "121212121212121212"
216 REM C(288) = "121212121212121212"
217 REM C(289) = "121212121212121212"
218 REM C(290) = "121212121212121212"
219 REM C(291) = "121212121212121212"
220 REM C(292) = "121212121212121212"
221 REM C(293) = "121212121212121212"
222 REM C(294) = "121212121212121212"
223 REM C(295) = "121212121212121212"
224 REM C(296) = "121212121212121212"
225 REM C(297) = "121212121212121212"
226 REM C(298) = "121212121212121212"
227 REM C(299) = "121212121212121212"
228 REM C(300) = "121212121212121212"
229 REM C(301) = "121212121212121212"
230 REM C(302) = "121212121212121212"
231 REM C(303) = "121212121212121212"
232 REM C(304) = "121212121212121212"
233 REM C(305) = "121212121212121212"
234 REM C(306) = "121212121212121212"
235 REM C(307) = "121212121212121212"
236 REM C(308) = "121212121212121212"
237 REM C(309) = "121212121212121212"
238 REM C(310) = "121212121212121212"
239 REM C(311) = "121212121212121212"
240 REM C(312) = "121212121212121212"
241 REM C(313) = "121212121212121212"
242 REM C(314) = "121212121212121212"
243 REM C(315) = "121212121212121212"
244 REM C(316) = "121212121212121212"
245 REM C(317) = "121212121212121212"
246 REM C(318) = "121212121212121212"
247 REM C(319) = "121212121212121212"
248 REM C(320) = "121212121212121212"
249 REM C(321) = "121212121212121212"
250 REM C(322) = "121212121212121212"
251 REM C(323) = "121212121212121212"
252 REM C(324) = "121212121212121212"
253 REM C(325) = "121212121212121212"
254 REM C(326) = "121212121212121212"
255 REM C(327) = "121212121212121212"
256 REM C(328) = "121212121212121212"
257 REM C(329) = "121212121212121212"
258 REM C(330) = "121212121212121212"
259 REM C(331) = "121212121212121212"
260 REM C(332) = "121212121212121212"
261 REM C(333) = "121212121212121212"
262 REM C(334) = "121212121212121212"
263 REM C(335) = "121212121212121212"
264 REM C(336) = "121212121212121212"
265 REM C(337) = "121212121212121212"
266 REM C(338) = "121212121212121212"
267 REM C(339) = "121212121212121212"
268 REM C(340) = "121212121212121212"
269 REM C(341) = "121212121212121212"
270 REM C(342) = "121212121212121212"
271 REM C(343) = "121212121212121212"
272 REM C(344) = "121212121212121212"
273 REM C(345) = "121212121212121212"
274 REM C(346) = "121212121212121212"
275 REM C(347) = "121212121212121212"
276 REM C(348) = "121212121212121212"
277 REM C(349) = "121212121212121212"
278 REM C(350) = "121212121212121212"
279 REM C(351) = "121212121212121212"
280 REM C(352) = "121212121212121212"
281 REM C(353) = "121212121212121212"
282 REM C(354) = "121212121212121212"
283 REM C(355) = "121212121212121212"
284 REM C(356) = "121212121212121212"
285 REM C(357) = "121212121212121212"
286 REM C(358) = "121212121212121212
```

Tri-oids

on BBC Micro

This game uses less than 4K bytes of memory on the Model B so will presumably run on the Model A if a different mode is used, and if the program is modified.

Tri-oids is similar to Asteroids. Although written in Basic — and therefore quite slow — the game has entertained my children and their school friends for hours during the holidays.

The object of the game is to hit any one of the three asteroids as many times as possible, without wasting shots. Hitting an asteroid scores up to 40 points depending on how far away it is, and gives you three extra shots.

For every 50 points scored an extra life is awarded. Lives are lost every time the asteroids come too close. On screen scoring is provided and so it is easy to see when to be careful. The controls are described in the program.

Program notes

Lines 29-30 Display title.
43-140 Instructions.
158 Envelope for laser.
169 Envelope for explosions.
178-249 Define characters for ship.
250 Define asteroid.
266 Dimension array for asteroid position: High
270 Set mode for game; join cursors.
289 Define graphics and text area.
294 Start positions for asteroids.
319 Set variables for new game.

- 328 Move asteroids: Cursor to ship position: print ship.
- 330 Check for game over.
- 343 Get key pressed.
- 358 Travel in right direction.
- 364 Fire laser in right direction.
- 378-380 Rotate right or left.
- 394 Hyperdrive.
- 10-544 Subroutines for new ship position.
- 50-622 Subroutines for laser direction and increment.
- 644 Calculate score and check if highest.
- 727 Display scores etc. (game over).
- 30-750 Play again or not?
- 75-866 Increment laser position/ checking for hit.
- 79-966 Update score; explosion sound; move to next position; check for hit and fire.
- 79-966 Also get new position for hit asteroid (900).
- 13-1390 Move asteroids and check if close to ship.
- 9-1990 Decrement lives; update score; do large laser sound.
- 9-1220 Draw or erase lines for explosions.
- 15-1950 Print score at top of screen.

```

510RETURN
520X%:=X%-20
530Y%:=Y%+20
540RETURN
550%:Z=0:Y%:=0:RETURN
560%:Z=0:Y%:=0:RETURN
570%:Z=0:Y%:=0:RETURN
580%:Z=0:Y%=-0:RETURN
590%:Z=0:Y%=-0:RETURN
600%:Z=-0:Y%=-0:RETURN
610%:Z=-0:Y%:=0:RETURN
620%:Z=-0:Y%:=0:RETURN
630MODE7
640HIGH=H%+H%*501 IFH%>HS HS=H%:F=1
650IFF=1 PRINTTAB(0,6)CHR$(136);"N E W . "
660PRINT"High Score ";HS
670PRINTTAB(14,10);CHR$(141);"Game Over."
680PRINTTAB(14,11);CHR$(141);"Game Over."
690PRINT" YOU SCORED ";HS;" HITS."
700PRINT"      WITH ";S%;" SHOTS LEFT"
710PRINT"      AND ";L%;" LIVES LEFT."
720PRINT" ANOTHER GAME (Y OR N) "
730A$=GET:IFA$=B9 GOTO270 ELSEIFA$>7B GOTO730
740CLS:PRINTTAB(16,14);CHR$(141);CHR$(136);"BYE... "
750PRINTTAB(16,15);CHR$(141);CHR$(136);"BYE... "
760END
770DEFPROCCLS:J=0:P=0
780X%:=X%-16:Y%:=Y%-16:XX:=X%:YY:=Y%
790HDX%,Y%
800REPEATJ=J+1:XX:=XX+X%:YY:=YY+Y%:IFPOINT(XX,Y%)=1 THENP
810DRAWX%,Y%
820UNTILJ=200R P<>0
830IFP<>0:PROCEX
840HDX%,YY:PLDT7,XX+J*X%,YY+J*Y%:HDX%,YY
850X%:=X%-16:Y%:=YY+16
860ENDPROC
870DEFPROC
880HIGH=H%+(P%2):S%:=S%+3:IFH%>49 H%:=H%-50:H%+1:L%:=X%+1

```

to next page

Open Forum

A GREAT NEW COMPETITION WORTH \$100,000 TO THE WINNER
Whizz-Kid '82

Fancy your chances?

We're looking for a bright young thing who can out-shine all the commercial software houses and come up with a sparkling new program that can be marketed commercially.

We want you to prove you can write a selling program and if you win the competition you'll be well on the way to making big money.

The winner will receive:

1. A Dragon 32 computer.
2. Advice from Popular Computing Weekly on how to market and sell the winning software and how to form and finance the company to do so.
3. £2,000-worth of free advertising in Popular Computing Weekly.

Entries to the award scheme must be accompanied by at least four out of five of the numbered coupons published in Popular Computing Weekly throughout September. The closing date for the competition is October 18. The winning entry will be announced in the issue published on November 18.

四

1. There is no limit on the number of entries you can send in, but each entry must be accompanied by four differently numbered competition coupons.
2. Closing date for entries is October 18, 1982.
3. The names of the winners will be announced in the November 18 issue of *Popular Computing Weekly*.
4. The Judges' decision is final.
5. No employees of Sunshine Publications Ltd. or their families, *plus their families*, in entries this competition.

The winner will be the author who submits the most commercially viable program together with a written outline of the author's own proposals on how he would run his software house and why he would like to do it. The judge will be *Popular Computing Weekly* editor, Brendon Gore.

*Popular Computing
Weekly*
Whizz-Kid '82 Scheme

Fill in this coupon. When you have collected four differently numbered coupons, send them with your program to: Popular Computing Weekly, Whizz-Kid '82, Houbhouse Court, 19 Whitcomb Street, London WC2.



If a number of equally good and commercially viable programs are submitted the decision of the overall winner will be based on the best accompanying written outline of the author's proposals for running a software house.

NAME: _____

2

Programming

Automatic relocation subroutine

Malcolm Peltz presents a relocation program for the expanded Vic20.

This subroutine (Fig 1) will allow a program on Vic20, with 8K or more expansion, to relocate itself automatically beyond the user defined character set. The subroutine is self-contained.

Line 10 checks to see if the program has already been relocated or if it has any expansion memory in which to relocate. If the program has already been relocated it will not repeat the process. Lines 20-40 work out the last address of the program, while line 50 uses this last address to test whether or not there is sufficient memory available to relocate the program.

Lines 60-70 then move the program backwards, to avoid overwriting itself.

Lines 80-120 re-adjust the links used by Vic Basic and determine where the variables may start. Lines 130-160 Poke these values into the appropriate locations and set the new start of Basic.

Line 170 then runs the relocated program, which starts at line 200. This time the test at line 10 will be true and the Goto taken. Line 170 could read RUN 200 if you wish.

READY.

```
1 PRINT "PLEASE WRITE!!"
5 POKE56,30:POKE52,30:POKE644,30:POKE44,16
6 POKE642,16:POKE4096,0:CLR
10 FORI=7680T08191:POKEI,32:POKEI+30720,1:NEXT
20 POKE210,30:FORI=2170T228:POKEI,158:NEXT
30 POKE229,159:POKE230,159:POKE231,31
40 FORI=232T0240:POKEI,159:NEXT
50 POKE244,151:POKE648,30
60 POKE36869,240:POKE36866,150
70 END
```

READY.

READY.

Fig 2

Contact wear problem solved

J S and J C Dale explain how to run machine code games on an expanded Vic20.

After buying an expansion unit and a 16K Ram pack for my Vic, I was very disappointed when I found that I could no longer run my machine code games. The only way to run these games was to remove the Ram pack, but this wears down the contacts.

Under great pressure from my children, my eldest son and I, armed with Nick Hampshire's excellent book *The Vic Revealed*, set about solving this problem. The accompanying listing (Fig 2) is the result.

Before any game written for an unexpanded Vic is loaded, this program should be loaded and run. The relevant program may then be loaded and run without removing the Ram pack.

If you want to write a program immediately after running the conversion program, type in *New* before starting. Otherwise a Syntax error will result.

Important — After using the unexpanded program, ensure that you reset the Vic by typing *SYS64802* then press *return*.

```
10 IFPEEK(44)=360RPEEK(56)=32THEN200
20 A=PEEK(43)+256*PEEK(44):LO=A
30 IFPEEK(A)=0ANDPEEK(A+1)=0THEN50
40 A=PEEK(A)+256*PEEK(A+1):GOTO30
50 HI=A+1:IFHI+4609>PEEK(55)+256*PEEK(56)THEN
PRINT"NOT ENOUGH ROOM TO MOVE THIS PROGRAM":END
60 FORA=HI+1TOLO-1STEP-1
70 POKEA+4608,PEEK(A):NEXT:A=LO
80 IFPEEK(A+4608)=0ANDPEEK(A+4609)=0THEN120
90 POKEA+4609,PEEK(A+4609)+18
100 X=PEEK(A+4608):Y=PEEK(A+4609)
110 A=PEEK(A+4608)+256*(PEEK(A+4609)-18):GOT080
120 X=X+2:IFX>255THENX=X-255:Y=Y+1
130 POKE47,X:POKE48,Y
140 POKE49,X:POKE50,Y
150 POKE45,X:POKE46,Y
160 POKE43,1:POKE44,36
170 RUN
200
```

Fig 1

Spectrum

In this new slot various contributors explore different aspects of the ZX Spectrum.

This program was designed to be used by schools or small businesses, to keep track of used stock. It also enables you to quickly discover which items are in short supply and therefore need to be reordered. The suppliers reference number is also listed, to make re-ordering even easier.

Items can be altered, entered or deleted at will, as there is a comprehensive selection. After being entered, stock items can be altered as they are used by inputting the number of items used preceded by a minus sign.

A copy of the current stock position or of the items that need to be re-ordered, can be obtained with a printer using Copy. When a list is required of the final few items, ie not a complete screenful, just press Break followed by Cts and Cont.

Tracking down the missing supplies

J Reynolds stock control program keeps track of your business.

On the 16K Spectrum this program will hold 125 separate items. If more items are required, you can use separate tapes for different types of stock. For example, stationery could be kept on one tape and painting materials on another. Sasco Sis-

index panels can be used to keep day to day track of stock levels so that information in the computer can be updated at weekly or monthly intervals.

In the event of any problems Goto 1000 will obtain the menu selection, so avoiding the loss of any variables which have been inputted.

Program notes

Line 19 sets up the arrays.
1919-1990: Menu selection.
1980-1988: This is the area of the programme dealing with the selection.
2000-2110: This part deals with alteration to the stock levels.
3000-3040: Enters the stock.
4000-4979: Lists the stock.
5000-5849: Lists items to be re-ordered.
6000: Saves and verifies.

Sound & vision

This program was written for the BBC micro, model A or B, and plays the musical game of Simon. The object of the game is to repeat the ever-increasing sequence of lights and tones by using the four cursor keys which correspond to the computer's coloured squares.

If you manage to reach your inputted skill level the computer will reward you with a warbling sound. Any mistakes and the game will end.

The program consists mainly of two For-next loops. The Y loop (line 200) controls the point in the sequence that you have successfully repeated. The X loop is used twice (line 210 and line 290) and runs from the beginning of the sequence to Y, controlling which light and tone is to be



Simon says . . .

L Raynor presents Simon, a musical game for the BBC model A or B.

made. The pattern of the lights in the sequence is stored in the A array which is dimensioned (line 170) to store as many lights as the skill level you had previously entered.

The lighting up effect is made by changing the appropriate colour, determined by the number (1-4) in the array, to white and then back to its original colour, by the use of the VDU 19 command (lines 230, 260, 390 and 420). The FX 4.2 command enables the cursor keys to be detected by an Inkey statement and the FX 15.0 command clears the sound buffer.

One useful procedure is the ProcWait() one, which pauses for T milliseconds. I managed to reach a score of 14 — can anyone beat that?

```
10 REM*** SIMON BY L.M. RAYNER ***
20 REM** FOR BBC MICRO MODEL A OR B ***
30 MODE 7
40 FOR X=1 TO 2:PRINTCHR$(141)CHR$(129)"           SIMON":NEXT X
50 PRINT""
60 PRINTCHR$(134)" REPEAT MY SEQUENCE OF SOUNDS AND"
70 PRINTCHR$(131)"COLCURS BY USING THE FOUR CURSOR KEYS"
80 PRINTCHR$(134)"WHICH CORRESPOND TO MY FOUR SQUARES."
90 INPUT"      SKILL? (10-30) "Z
100 IF Z<10 OR Z>30 THEN 90
110 *FX 4,2
120 MODE 5
130 FOR X=0 TO 3:VDU 19,X,:+1,0,0,0:NEXT X
140 GCOL0,1:MOVE 640,512:MOVE 0,512:PL0T85,640,1 24:PL0T 85,0,1024
150 GCOL0,2:MOVE 640,512:MOVE 640,1024:PL0T 85,1280,512:PL0T 85,1280,
1024
160 GCOL0,3:MOVE 640,512:MOVE 640,0:PL0T 85,1280,512:PL0T 85,1280,0
170 DIM A(2)
180 FOR X=1 TO Z:A(X)=RND(4)-1:      30:GOTO 510
190 NEXT X
200 PROCWAIT(100)
210 FOR Y=1 TO Z
210 FOR X=1 TO Y
220 SOUND 1,-15,A(X)*20,30
230 VDU 19,A(X),7,0,0,0
240 PROCWAIT(50)
250 *FX 15,0
260 VDU 19,A(X),A(X)+1,0  +0
270 PROCWAIT(50)
280 NEXT X
290 FOR X=1 TO Y
300 A$=GETS
310 A=9
320 IF A$=CHR$(139) THEN A=0
330 IF A$=CHR$(136) THEN A=1
340 IF A$=CHR$(137) THEN A=2
350 IF A$=CHR$(138) THEN A=3
360 IF A=9 THEN 30
370 IF A(X)<>A THEN SOUND 1,-15,1, 570 ENDPROC
380 SOUND 1,-15,A(X)*20,30
390 VDU 19,A(X),7,0,0,0
400 PROCWAIT(50)
410 *FX 15,0
420 VDU 19,A(X),A(Y)+1,0,0,-1
430 PROCWAIT(50)
440 NEXT X
450 PROCWAIT(120)
460 NEXT Y
470 FOR R=1 TO 20
480 SOUND 1,-15,150,2
490 SOUND 1,-15,40,2
500 NEXT R
510 COLCUR 2:PRINT TAB(5,10)
510 "SCORE=";Y-1
520 PRINT"**PRES SPACE TO START"
530 A$=GETS:R=IN
540 DEPROCWAIT(T)
550 TIME=0
560 REPEAT UNTIL TIME=
```

Peek & poke

Peek your problems to our address. Ian Beardmore will poke back an answer.

ABOMINABLE CURSORS

K V Jefferies of East Walk, Croydon Road, Reigate, writes:

Q After using my ZX81 for seven months I am having difficulty in switching from 'K' cursor to 'F' cursor. With the Shift key depressed, pressure on New line does not immediately produce the Function cursor. Is there any simple solution?

I was thinking of buying a Kempston keyboard. Will this solve the problem, or will it be a waste of money, if the touch sensitive system is faulty?

A I have a big sticker on my ZX81. 'New line worn. Roll the finger over the key to make contact.' As your machine is within the guarantee period, in theory you should send it back. If you do, test it as soon as Sinclair return it to you. If there is no improvement send it back on the same day, with a stiff covering letter.

Some people who have had experience of the Sinclair returns department would advise you very strongly against doing this if you want to see your machine back in anything like a reasonable time.

There is not a lot of evidence of ZX81 keyboards giving out, so this case should be seen in isolation. If the keyboard is at fault, then another keyboard should bypass the problem.

The only exception would be if one of the diodes had become damaged in some way, preventing it from discharging power at the right time. This is however very unlikely.

EACH ONE TO HIS TASTE

Michael Andrews of South East London, writes:

Q I have just been given a Texas Instruments 99/4 computer for a birthday present. Could you please tell me if there are any books or maga-

zines published for this computer, or if there is a user's club? It seems that it is not a very popular computer and does not have much support.

A I do not know about horses, but perhaps you are looking a gift computer a little in the mouth. Think of all those BBC and Sinclair customers who have been waiting literally months for computers that still have not been delivered. While the TI 99/4 might not be well supported, it is not without admirers. It has a home users group who should be able to give you the sort of information you need. Write to Paul Dicks, TIHOME, 157 Bedford Road, Morden, Surrey.

COMMANDING PROBLEMS

Stephen Clements of Hophurst Drive, Crawley Down, West Sussex, writes:

Q I am writing to ask if you could help me out with a problem I have been agonising over for the past few weeks. I have been converting programs for the ZX81 to be used on my Vic, but I have come face to face with a line *Print at x,y*; which I do not know how to convert. Can you help me?

A One of the major disadvantages of the ZX81 is that it does not have a memory mapped screen. Therefore you cannot *Poke* characters on to the display, though you can use the *Print at* command. *Print at x,y; "*" would mean that at line x, column y, the character ** would appear.

But the ZX81 can *Print at* more screen locations than the Vic20 can *Poke*. This is because the ZX81 uses a 22 line \times 32 column grid, whereas the Vic uses a 24 line \times 22 column grid.

The control of characters will have to be different, because a location defined by a *Print at* command is not mapped into a specific place in the memory. Such a location will float in the Ram, without a permanent address that can be *Peeked* or *Poked*.

Horizontal locations are determined by adding or subtracting from y, while vertical locations are determined by adding or subtracting from x. A simple routine to show this would be:

```
10 LET X=10
20 LET Y=10
30 PRINT AT X,Y; "*"
40 IF NOT INKEY$= " " THEN PRINT
   AT X,Y; " " a space
50 IF INKEY$="S" THEN LET
   Y=Y-1
60 IF INKEY$="E" THEN LET
   X=X+1
70 IF INKEY$="T" THEN LET
   X=X-1
80 IF INKEY$="B" THEN LET
   Y=Y+1
90 GOTO 30
```

PACKAGING MEMORIES

M Ellick of Burington Close, Bristol, writes:

Q I have a ZX81 with a 64K Ram Pack fitted. In a letter in your July 1 issue, you mention that the ZX Microdrive will probably be able to be used with other computers, like the ZX printer. Do you think that an adaptor will be needed to use the Microdrive with the ZX81?

A Until someone actually sees the Microdrive we cannot be sure how easy it will be to use with the ZX81. However, the ZX81 does not have the disc control commands in its Rom, so some sort of extra control will be necessary. My guess is that this will take the form of an Eprom accessible by *Poke* commands. This means that you will have to switch out another block of your memory pack to make way for the Eprom. But, this would still allow you to write a 48K program, with a database stored on the Microdrive.

SWITCHING BLACK AND WHITE

Imtiaz Mirza of Priory Avenue, High Wycombe, Buckinghamshire, writes:

Q I read in PCW July 8 that it was not possible to modify the ZX81 hardware to change the black characters on a white background to white

characters on a black background. You said that this was only possible on the ZX80. Could you please advise me on how to do this?

A The modification on the ZX80 to give inverse video is a simple wiring job. On the underside of the PCB are three letters A, B and C. Normally, the track runs from A to C. A simple wire bridge between A and B, instead of A and C, will do the job. If you want to get really clever, you could always wire in a switch, so that you could change between an ordinary and inverse display.

D Fritsch, of 6 Stanton Road, Felwall, Warrington, Cheshire WA4 2HS, makes an add on that will give you an inverse display on the ZX81.

HOODWINKING KEYWORD MEMORY

Alan Hapgood of Fulham, London, writes:

Q I know that if I want to use a keyword on my Vic20, I can enter it by using the first letter, followed by the second letter shifted. What I would like to know is why this is so?

A This useful feature on the Vic is due to the operating system which uses text compression. All the keywords on a Vic are stored as a number between 0 and 255. If you write *Input* on to the screen, it will be stored there as five bytes, and will therefore take up five bytes of screen memory. But it will leave the buffer as a single byte with a decimal value of 132. Whatever the number of bytes of screen memory the keyword takes, it will still only be stored as a single byte.

When you use an *I Shift N* for *Input* you are fooling the computer into thinking that you have entered the full command. This only uses two bytes of screen memory. If you *List* the instruction, it will come up as *Input*, because the process is reversed as information comes back to the screen out of the buffer.

POPULAR Computing WEEKLY

HOBHOUSE COURT, 19 WHITCOMB STREET WC2

WHY NOT TAKE OUT A SUBSCRIPTION?

You can have *Popular Computing Weekly* sent to your home for £19.95 a year. This will cover the cost of the magazine, postage and packing.

Fill in the coupon below and keep yourself up to date week by week.

I would like to take out subscription(s) to *Popular Computing Weekly*.

Name.....

Address.....

Tel No.

I enclose £..... for subscription(s).

Please make cheques payable to Sunshine Publications Ltd.

Classified

Do you want to sell your computer and buy a bigger and better one?

Have you ever thought of trying to make some money out of selling tapes of your own programs?

Whatever it is you want to buy or sell why not use our classified pages? It has to be better than waiting for up to nine weeks to get into one of the old monthly magazines.

Not only that, but our rates are very reasonable.

For private individuals it only costs 20p per word, with a minimum of 10 words. We can make it so cheap because we charge companies using the classified columns 40p per word.

The classified pages can be used for semi-display advertising. The cost for this is £10 per single column centimetre, with a minimum charge of £30.

All copy for the classified pages must be pre-paid. Cheques/postal orders should be made out to *Popular Computing Weekly*, and your advertisement should arrive at least two weeks before the publication date.

For semi-display advertising please call David Lake on 01-839 2846.

For classified entries please fill in the form below and send it, with your cheque/postal order, to Classified Department, *Popular Computing Weekly*, Hobhouse Court, 19 Whitcomb Street, London WC2.

If you have any queries regarding classified advertising please call Alastair Macintosh on 01-930 3840.

Please write your copy in bold capital letters on the lines below.

Here's my classified ad.

I make this words, at per word so I owe you £

Name.....

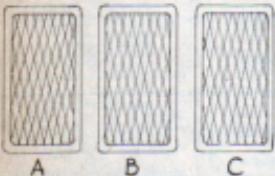
Address.....

Telephone.....

Please cut out and send this form to:
Classified Department, *Popular Computing Weekly*, Hobhouse Court, 19 Whitcomb Street, London WC2

Competitions

Finding the lady



Three playing cards are placed face down on a table. The total value of A and B is 15. The total value of B and C is 17. No card is a seven or has a value higher than 9. Identify the cards.

This problem can be solved using a simple program:

```
10 FOR A = 1 TO 9
20 LET B = 15 - A
30 LET C = 17 - B
40 IF A = 7 OR A > 9 OR B = 7 OR B > 9 OR C = 7
    OR C > 9 THEN GOTO 80
50 PRINT "A = "; A; "B = "; B; "C = "; C
60 NEXT A
```

Even without a computer this is not a difficult problem, but the same approach will apply to more complex problems.

Now try this card trick. Deal 20 cards off a shuffled pack and ask a friend to insert the cards back into the pack, but the other way up from the rest of the pack (face up rather than face down). Shuffle the pack and get your friend to count off the top 20 cards, straightening the pile so that you cannot see which way they face. Ask

your friend to note, without telling you, the number of cards face-up in the remaining 32 cards.

Hold the 20 cards behind your back and — by thought transference — you will be able to turn individual cards over so that the number of cards face-up in your part of the pack is the same as the number face-up in your friend's 32 cards.

When the two piles of cards are investigated, they will be found to have the same number of cards face-up.

How is it done? When you hold the cards behind your back you simply turn them over.

The whole pack contains 52 cards, of which 20 have been turned over. If your friend has a number of cards, p , turned over in his pile of 32 cards then you have $20 - p$ reversed cards in yours. Since there are 20 cards in all in your pile then you must have p cards face down. Turn the whole pile over and you have the same number of cards face-up as your friend.

shuffling until the pack is brought back to its order at the start.

How many times must the shuffle be performed if (a) a standard deck of 52 cards is used and (b) if the jokers are added to make a pack of 54 cards?

Solution to Puzzle No 17

First find the length of the sides of a non right-angle triangle where the area is 180 sq miles and the sides are exact numbers of miles in length. This can be done using a program similar to the one below.

```
10 LET L = 10
20 FOR A = L - 9 TO L
30 FOR B = 2 TO A - 1
40 FOR C = (A - B) + 1 TO B - 1
50 LET S = (A + B + C) / 2
60 LET T = SQR (S * (S - A) * (S - B) * (S - C))
70 IF ABS (T - 180) < 0.000001 AND B + B + C + C < > A + A THEN GOTO 290
80 NEXT C
90 NEXT B
100 NEXT A
110 LET L = L + 10
120 GOTO 20
200 PRINT A; ":"; B; ":"; C
210 STOP
```

This gives the sides as 37, 30 and 13 miles. The catch was in the words "the morning's total divided by half" — meaning multiplied by two. In this case the distance from A to B could only be 13 miles — covered in the morning — with 26 miles covered in the afternoon. With a total round trip of 80 miles, this leaves 41 miles still to be travelled to finish the journey.

Winner of Puzzle No 17

The winner is P M Hill, Westcott Close, Frome, Somerset who receives £10.

ARTHUR MEETS A FOREIGNER, CALLED ALAN OG

YOU ARE SO ABSOLUTE, ARTHUR, IN THE END IT'S YES OR NO, IT'S EITHER FALSE OR TRUE; WHO'S NOT YOUR ENEMY MUST BE YOUR FRIEND; IT ALL COMES DOWN TO DAY OR OFF WITH YOU.

NOW LOOK AT ME, I RISE AND FALL, RECEDING APPROACH APPROPRIATE AND MERGE, I USE MY JUDGEMENT, DIVE, FLOW, FOLLOW, VARY SPEED, I NEVER JUMP, REJECT, EXCLUDE, OR CHOOSE.

AND MOVERS LOVE ME, I'M LIKE ONE OF THEM, I ASK NO QUESTIONS, SO THEY TELL NO LIES, SOON THEY'LL FORGET THEY MADE ME WHAT I AM.

THEY HATE YOU, ARTHUR, FACE THE FACT THEY HATE YOU.

YES, THEY ADMIRE YOU: YOU CAN ANALYSE, BUT ME — I HOLD THE MIRROR UP TO NATURE.



Lawrence Lerner & James Macdonald
Published by THE INTERADVERT PRESS LTD, 1497 STREET,
DUBLIN 7. TEL. 01-763 1039. TELEX 2140.

Probably the fastest microcomputer in the universe

the **JUPITER ACE** only £89.95.



Key Features

- Revolutionary microcomputer language FORTH.
- Full-size moving-key keyboard.
- User-defined high-resolution graphics.
- Programmable sound generator.
- Floating point arithmetic.
- Fast cassette interface.
- Upper and lower case ASCII character set.
- 24 x 32 character flicker-free display.

The Jupiter Ace uses FORTH

The Ace is set apart from all other personal computers on the market by its use of a revolutionary language called 'FORTH'. Some computer languages are easy for humans to understand, others are easy for computers; FORTH is most unusual in being both. Its underlying principles are so simple that it takes even a newcomer to computers only a few minutes to learn how to do calculations on the Ace, yet the very same principles are powerful enough to allow you to invent your own extensions to the language itself.

At the same time, the memory-saving coded form used to store your programs inside the Ace allows it to obey them very fast — typically in less than a tenth of the time it would take to do the same thing using a different language. Amongst other things, this makes the Ace ideal for games.

FORTH's unique combination of speed, versatility and ease of programming has already made it a prime choice for professional applications as diverse as pub games and radio telescopes, and gained it an enthusiastic national user group. Now the Jupiter Ace can bring this addictive language into your own home.

Designed by Jupiter Cantab

Leading computer Designers Richard Altwasser and Steven Vickers have a reputation for pushing technology forwards. After playing the major role in creating the ZX Spectrum they formed Jupiter Cantab to develop their latest brainchild the Jupiter Ace.

All inclusive Price

For £89.95 you receive your Jupiter Ace, a mains adaptor, all the leads needed to connect to most cassette recorders and T.V.s (colour or black and white), a software catalogue and a manual.

The manual is a complete introduction to the world of personal computing and a course in FORTH programming on the Ace.

Even if you are a complete newcomer to computers, the manual will guide you step by step from first principles to confident programming.

The price includes postage packing and V.A.T.

Technical Specification

Hardware

Processor/Memory

Z80A running at 3.5 MHz.
8K bytes ROM 3K bytes RAM.

Input

40 moving-key keyboard with auto-repeat on every key.

Output

Memory-mapped 32 x 24 character display with high resolution user graphics. Output to drive normal UHF TV set or channel 36.

Sound

Provided by internal loudspeaker.

Cassette

Load Save & Verify at 1500 baud, separate data storage.

Software, FORTH

Data Structures

Integer, Floating point and String data may be held as constants, variables or arrays with multiple dimensions and mixed data types.

Control Structures

IF-THEN-ELSE, DO-LOOP, BEGIN- WHILE-REPEAT, BEGIN-UNTIL, all may be mixed and nested to any depth.

Operators

Mathematical +, -, X, =, Logical AND, OR, NOT, XOR.

Comparison <, >, =.

Program Editing

FORTH words may be listed, edited and redefined. Comments are preserved when words are compiled.

Order Form

The Jupiter Ace is available only by mail order. Please allow up to 28 days for delivery.

Send cheque or postal order with the form to:—

JUPITER CANTAB, 22 FOXHOLLOW, BAR HILL, CAMBRIDGE CB3 8EP

Please send me:—

JUPITER ACE MICROCOMPUTER(S) @ £89.95.

Name: Mr/Mrs/Miss

Address

ZX81..SPECTRUM..B.B.C. MICRO..ATOM..VIC

NEW!

ASPECT Assembler

FOR

**SINCLAIR
ZX Spectrum**

NOW! unleash the full power of the Z80 micro-processor in your incredible Spectrum, and program for yourself those fast-moving machine code games. Our unique 'Aspect' Assembler has a full screen Editor, and the Assembler can generate code

for any address. It has a built-in, foolproof, error-detection system, comes complete with full instruction manual, and will run on 16K and 48K Spectrum.

Just
£9
inc. post &
packaging

Spectral INVADERS

The original and the best! The very first, arcade-quality machine-code game to appear on the market for the sensational ZX Spectrum.

Just
£5
inc. post &
packaging

IMMEDIATE DELIVERY

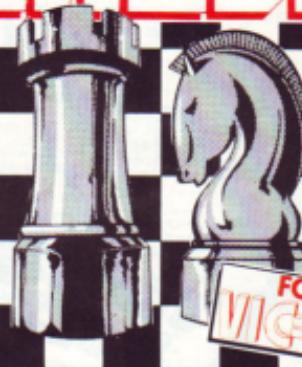
Send stamped addressed envelope for details of Z8081 and Atom Software

BUG-BYTE SOFTWARE

BUG-BYTE SOFTWARE. FREEPOST (No stamp req.) LIVERPOOL L3 3AB

NEW!

CHESS



**FOR
VIC 20**

Hot on the heels of our quality chess programs for the BBC and Atom Micros, we introduce, for the first time in the U.K., chess on your VIC! (16K expansion). Once again there are 1000 levels of play, plus many options, chess problems, game storage etc. Deals with castling and en passant.

Just
£7
inc. post &
packaging

BBC CHESS £11.50
ATOM CHESS £9.00

... also for your

VIC

Another VIC in the wall
VIC Gammon
VIC Panic
VIC Cosmoids
VIC Asteroids

ALL PRICES INCLUDE POSTAGE

ALL
£7
inc. post &
packaging

Please send me _____
Enclosed cheque/P.O. for _____

OR Please debit my Access 5224 _____
BARCLAYCARD 4829 _____ Expiry date _____

Name _____
Address _____

Dealers Discount Available
ACCESS BARCLAYCARD ORDERS WELCOME ON 24hr
ANSAPHONE 051 227 2642 or send to

BUG-BYTE SOFTWARE
FREEPOST (No stamp req.)
LIVERPOOL L3 3AB

POPCW 9.82



ZX81..SPECTRUM..B.B.C. MICRO..ATOM..VIC

